

Документ подписан простой электронной подписью

Информация о владельце:

ФИО: Порохня Андрей Александрович

Должность: И.о. директора института перспективной инженерии

Дата подписания: 18.06.2026 14:18:48

Уникальный программный ключ:

990bea3aeec48c23bd8699e48288f8d1960c800

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение
высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

УТВЕРЖДАЮ

И.о. директора института
перспективной инженерии,
кандидат технических наук,
доцент

А.А. Порохня

**ФОНД ОЦЕНОЧНЫХ СРЕДСТВ ПО ДИСЦИПЛИНЕ
№SQL СУБД**

Направление подготовки	09.03.02 Информационные системы и технологии
Направленность (профиль)	Прикладное программирование в интеллектуальных информационных системах
Форма обучения	очная
Год начала обучения	2026
Реализуется в 8 семестре	

Введение

1. Назначение: Оценочные материалы (оценочные средства) предназначены для проведения текущего контроля успеваемости и промежуточной аттестации обучающихся, и представляют собой совокупность контрольно-измерительных материалов и методов их использования для измерения уровня достижения обучающимся установленных результатов обучения.

2. ФОС является приложением к программе дисциплины «NoSQL СУБД»

3. Разработчик Николаев Е.И., доцент департамента цифровых, робототехнических систем и электроники

4. Проведена экспертиза ФОС.

Члены экспертной группы:

Председатель: Дроздова В. И., профессор департамента цифровых, робототехнических систем и электроники

Члены комиссии:

Краюткина Е.В., доцент департамента цифровых, робототехнических систем и электроники

Николаев Е.И., доцент департамента цифровых, робототехнических систем и электроники

Представитель организации-работодателя: А. В. Новиков, начальник отдела АСУТП ООО «ЕНДС-Ставрополь»

Экспертное заключение: ФОС по дисциплине «NoSQL СУБД» позволяет оценить уровень сформированности компетенций. Рекомендовать к использованию в учебном процессе.

5. Срок действия ФОС определяется сроком реализации образовательной программы.

1. Описание показателей и критериев оценивания на различных этапах их формирования, описание шкал оценивания

Уровни сформированности компетенци(ий), индикатора (ов)	Дескрипторы			
	Минимальный уровень не достигнут (Неудовлетворительно) 2 балла	Минимальный уровень (удовлетворительно) 3 балла	Средний уровень (хорошо) 4 балла	Высокий уровень (отлично) 5 баллов
Компетенция: ПК-6				
Результаты обучения по дисциплине (модулю): Демонстрирует знания в области архитектуры систем управления базами данных, типологии баз данных, методов оптимизации баз данных <i>Индикатор:</i> ПК-6ид-6.1	Обладает базовыми знаниями в области архитектуры систем управления базами данных (СУБД), типологии баз данных и методов оптимизации баз данных. Способен объяснить основные компоненты архитектуры СУБД, такие как клиент-серверная модель, система управления транзакциями, язык запросов. Знаком с основными типами баз данных, такими как реляционные, иерархические и сетевые, и может описать их особенности и преимущества.	Имеет глубокие знания в области архитектуры СУБД, типологии баз данных и методов оптимизации баз данных. Способен объяснить сложные концепции, такие как многоуровневая архитектура СУБД, применение хранилищ данных и репликация данных. Способен анализировать источники проблем производительности баз данных и предлагать методы их решения, такие как оптимизация запросов, использование индексов и денормализация.	Обладает экспертными знаниями в области архитектуры СУБД, типологии баз данных и методов оптимизации баз данных. Способен проектировать и реализовывать сложные архитектуры СУБД, включая кластеризацию, распределенные системы и резервное копирование данных. Способен использовать продвинутые методы оптимизации баз данных, такие как горизонтальное и вертикальное шардинг, разделение данных и партиционирование. Способен исследовать и внедрять новые технологии и инструменты, связанные с архитектурой СУБД.	Имеет глубокие и широкие знания в области архитектуры СУБД, типологии баз данных и методов оптимизации баз данных. Способен консультировать и руководить проектами по разработке и внедрению архитектуры СУБД, обеспечивая высокую производительность и надежность системы. Способен анализировать и решать сложные проблемы производительности баз данных, оптимизируя запросы, настраивая индексы и используя продвинутые техники кеширования. Способен разрабатывать стратегии масштабирования баз данных, управлять растущими объемами данных и обеспечивать безопасность и целостность данных.
Результаты обучения по дисциплине (модулю): Применяет инструментальные средства для проектирования	Демонстрирует знания основных понятий и принципами проектирования баз данных, включая распределенные базы данных, не имеет	Имеет опыт работы с инструментальными средствами для проектирования баз данных, включая распределенные.	Обладает критическим мышлением и глубокими знаниями в области проектирования баз данных различного типа.	Демонстрирует глубокие знания и опыт работы с инструментальными средствами для проектирования баз данных

баз данных различного типа, в том числе распределенных <i>Индикатор:</i> ПК-бид-6.2	практического опыта работы с инструментальными средствами для проектирования. Знает основные понятия и структуры баз данных, понимает пользу от распределенной базы данных, но не способен применять инструментальные средства для ее проектирования.	Способен использовать эти средства для создания баз данных различного типа и решения конкретных задач в рамках проектирования распределенных баз данных. Демонстрирует знание методик проектирования, способен создавать и оптимизировать схемы баз данных, управлять данными и обеспечивать их целостность.	Способен использовать инструменты и технологии, такие как NoSQL или реляционные базы данных, для работы с большими объемами данных и распределенной системой хранения данных. Способен применять передовые методы и подходы, такие как горизонтальное масштабирование или репликация, для создания и управления распределенными базами данных.	различного типа, включая распределенные. Способен проектировать сложные системы баз данных и создавать оптимальные стратегии для их распределения и репликации. Способен решать сложные задачи, связанные с проектированием и оптимизацией баз данных, а также разрабатывать новые инновационные подходы и методы в области проектирования баз данных
Результаты обучения по дисциплине (модулю): Владеет методикой оптимизации систем хранения данных на основе интерпретируемых критериев эффективности <i>Индикатор:</i> ПК-бид-6.3	Знаком с основными принципами и методами оптимизации систем хранения данных. Знает интерпретируемые критерии эффективности и умеет объяснить их принципы. Использует готовые методики оптимизации, разработанные другими специалистами.	Обладает достаточными знаниями и опытом в оптимизации систем хранения данных. Способен самостоятельно анализировать требования и задачи связанные с системами хранения данных и определять целесообразность использования интерпретируемых критериев эффективности. Пользуется несколькими методиками оптимизации и способен выбрать наиболее подходящую для конкретной ситуации.	Обладает глубокими знаниями в области оптимизации систем хранения данных и владеет несколькими методиками оптимизации. Способен анализировать и оптимизировать сложные системы хранения данных с использованием интерпретируемых критериев эффективности. Способен предлагать свои собственные методики оптимизации и модифицировать существующие.	Демонстрирует экспертные знания в области оптимизации систем хранения данных с использованием интерпретируемых критериев эффективности. Владеет всеми существующими методиками оптимизации и может разрабатывать свои собственные. Способен эффективно оптимизировать любые системы хранения данных и применять передовые технологии в этой области.
Результаты обучения по дисциплине (модулю): Анализирует предметную область в рамках решения профессиональных задач и проекти-	Понимает основные принципы анализа предметной области и проектирования архитектуры систем обработки данных. Может проводить базовый анализ исходных данных и	Обладает углубленным пониманием анализа предметной области и проектирования архитектуры систем обработки данных. Может выполнять	Обладает глубоким знанием анализа предметной области и проектирования архитектуры систем обработки данных. Может проводить комплекс-	Является экспертом в области анализа предметной области и проектирования архитектуры систем обработки данных. Может проводить слож-

рует архитектуру систем обработки данных на основании типов данных, конфигурации потоков данных <i>Индикатор:</i> ПК-бид-6.4	понимать их структуру и связи. Умеет выбирать подходящие типы данных для решения задач и оптимизировать потоки данных. Может создавать простые архитектуры для систем обработки данных.	сложный анализ данных и рассматривать их взаимосвязь и влияние на проектируемую систему. Умеет использовать различные типы данных и конфигурации потоков данных для оптимизации работы системы обработки данных. Может проектировать сложные архитектуры систем обработки данных и внедрять инновационные подходы.	ный анализ данных, генерировать новые типы данных и выстраивать сложные конфигурации потоков данных. Умеет использовать передовые технологии и инструменты для обработки данных, такие как машинное обучение и искусственный интеллект. Может проектировать инновационные архитектуры систем обработки данных, учитывая потребности и требования бизнеса.	ный исследовательский анализ данных и создавать уникальные типы данных и конфигурации потоков данных. Умеет применять передовые методы и алгоритмы для обработки данных, учитывая особенности предметной области. Может разрабатывать и внедрять инновационные решения в области обработки данных и аналитики.
--	--	---	--	---

ОЦЕНОЧНЫЕ СРЕДСТВА ДЛЯ ПРОВЕРКИ УРОВНЯ СФОРМИРОВАННОСТИ КОМПЕТЕНЦИЙ

Номер задания	Правильный ответ	Содержание вопроса	Компетенция
		Вопросы к экзамену:	
1.		1. Разработка приложений для MongoDB 2. Паттерны проектирования 3. Двоичные данные и GridFS 4. MongoDB на PHP, Java и C++ 5. Пространственные индексы 6. Развитие вычислительных технологий и стратегии их реализации 7. Зависимые и независимые витрины данных 8. Инкрементальный подход 9. Стратегии реализации 10. Инструментарий интеллектуального анализа данных 11. Базы данных NoSQL 12. Nadoop. История развития, особенности и инструменты 13. Интернет вещей 14. NoSQL и Big Data 15. История возникновения нереляционных подходов 16. Базовые принципы Big Data 17. Масштабируемость 18. Современное состояние Big Data 19. Правила 5V 20. Применение Big Data 21. Технологии Big Data 22. Последовательное чтение и прямой доступ 23. Модель программирования Big Data 24. Массивно параллельные вычислительные системы (MPP) 25. In-Memory базы данных 26. Системы MapReduce 27. BSP-системы 28. Системы Big Data и системы, основанные на транзакциях 29. Хранилище на основе упорядоченных столбцов 30. Хранилища ключ/значение 31. Документно-ориентированные базы данных 32. Графовые базы данных 33. Столбцовые наборы данных и РСУБД 34. Наборы столбцов, встроенные отображения, пары ключ/значение	ПК-6

		35. Webtable 36. Архитектура распределенного хранилища HBase 37. Документо-ориентированное хранилище 38. Memory-Mapped файлы 39. Коллекции и индексы MongoDB 40. Горизонтальное масштабирование 41. Хранилище типа ключ/значение в Memcached и Redis 42. Частично согласованные нереляционные базы данных 43. Элементы кластера Hadoop 44. Дистрибутивы и экосистема Hadoop 45. Apache 46. Cloudera 47. Hortonworks 48. MapR 49. Компоненты YARN 50. YARN Framework 51. Приложения YARN 52. Недостатки YARN 53. YARN и MapReduce 54. Процесс взаимодействия с YARN 55. Свойства конфигурации YARN 56. Применения YARN 57. Обработка графов 58. Обработка данных в режиме реального времени 59. Технология BSP 60. Стратегии объединения	
		Задания закрытого типа	
2.		Какие СУБД являются реляционными? +: MS SQL Server -: MongoDB +: MySQL +: PostgreSQL	ПК-6
3.		Какие СУБД являются документоориентированными? +: MongoDB -: MySQL -: PostgreSQL -: MS SQL Server	ПК-6
4.		В каком формате представляются документы в MongoDB?	ПК-6

		+: JSON -: XML -: XLS -: HTTP	
5.		Как происходит создание базы данных при использовании консоли MongoDB? +: база данных создается при вызове функции записи документа (документов), вместе с коллекцией, содержащей документы -: база данных создается специальной командой; MongoDB не предполагает создания базы данных «на лету» -: создать базу данных можно только из клиентского приложения на языке Python	ПК-6
6.		При описании подхода NoSQL используют правило 5V. Какие свойства присутствуют в правиле 5V? +: скорость (velocity) +: объем (volume) -: согласованность (consistency) -: атомарность (atomicity)	ПК-6
7.		При описании подхода NoSQL используют правило 5V. Какие свойства присутствуют в правиле 5V? +: достоверность (veracity) +: объем (volume) -: согласованность (consistency) +: значимость (value)	ПК-6
8.		Какое утверждение о формате JSON корректно? +: это текстовый формат -: это бинарный формат -: это табличный формат -: это мультимедиа-контент	ПК-6
9.		Какое утверждение об объекте JSON корректно? +: это неупорядоченное множество пар имя/значение, заключённое в фигурные скобки { } -: это неупорядоченное множество пар имя/значение, заключённое в квадратные скобки [] -: это класс, определяемый на языке Javascript	ПК-6
10.		Какой из объектов JSON определен корректно? +: {“name” : “Nik Corner”, “salary” : 2500, “address” : “No data”}	ПК-6

		-: /*"name" : "Nik Corner", "salary" : 2500, "address" : "No data"*/ -: [{"name" : "Nik Corner", "salary" : 2500, "address" : "No data"}] -: class P{name" : "Nik Corner", "salary" : 2500, "address" : "No data"} -: это класс, определяемый на языке Javascript	
11.		Какой из объектов JSON определен корректно? +: {"name" : "Nik Corner", "salary" : 2500, "address" : "No data"} -: <--"name" : "Nik Corner", "salary" : 2500, "address" : "No data"--!> -: [{"name" : "Nik Corner", "salary" : 2500, "address" : "No data"}] -: class P{name" : "Nik Corner", "salary" : 2500, "address" : "No data"}	ПК-6
12.		В консоли MongoDB необходимо выбрать базу users в качестве текущей. Какую команду консоли следует использовать? +: use users -: use database users -: using users -: select users	ПК-6
13.		какая команда консоли позволяет разработчику получить справку по функционалу консоли MongoDB? +: help -: use -: select -: what	ПК-6
14.		В консоли MongoDB требуется узнать имя текущей используемой базы данных. Какую команду надо использовать для этого? +: db.getName() -: db.help() -: db.use() -: db.select()	ПК-6
15.		Какая команда консоли MongoDB позволяет вывести список баз данных? +: show dbs -: dbs list -: list -: print databases	ПК-6
16.		Какая команда консоли MongoDB позволяет вывести список коллекций текущей базы данных?	ПК-6

		+: show collections -: print collections -: collections -: db.list()	
17.		Какая команда используется для запуска консоли MongoDB? +: mongo -: startmongo -: mongodb -: start mongodb	ПК-6
18.		Какая команда используется для импорта данных из внешнего источника в базу данных MongoDB? +: mongoimport -: importdata -: importmongo -: dataimport -: import	ПК-6
19.		Какие стратегии для работы с MongoDB доступны разработчику? +: работа с MongoDB посредством консоли +: работа с MongoDB с использованием специального графического приложения Compass -: работа с MongoDB из текстового редактора	ПК-6
20.		Необходимо добавить документ в коллекцию persons текущей базы данных. Какая команда является корректной? +: db.persons.insert ({“name” : “No name”}) -: db.persons ({“name” : “No name”}) -: db.persons.create ({“name” : “No name”}) -: db.insert ({“name” : “No name”})	ПК-6
21.		Необходимо добавить один документ в коллекцию persons текущей базы данных. Какие команды являются корректными? +: db.persons.insert ({“name” : “No name”}) -: db.persons ({“name” : “No name”}) -: db.persons.insertOne ({“name” : “No name”}) -: db.insert ({“name” : “No name”})	ПК-6
22.		Необходимо добавить один документ в коллекцию persons текущей базы данных. Какие коман-	ПК-6

		ды являются корректными? <pre> +: db.persons.insert({"name" : "No name"}) -: db.persons({"name" : "No name"}) -: db.persons.set({"name" : "No name"}) -: db.insert({"name" : "No name"}) </pre>	
23.		Необходимо добавить несколько документов в коллекцию persons текущей базы данных. Какие команды являются корректными? <pre> +: db.persons.insert ([{"name" : "No name"}, {"name" : "Max"}]) -: db.persons.select ([{"name" : "No name"}, {"name" : "Max"}]) +: db.persons.insertMany ([{"name" : "No name"}, {"name" : "Max"}]) -: db.persons.insert ({"name" : "No name"}, {"name" : "Max"}) </pre>	ПК-6
24.		Необходимо выбрать все документы из коллекции zoo текущей базы данных. Какую команду консоли следует использовать? <pre> +: db.zoo.find() -: db.zoo.select() -: db.zoo.findAndSelect() -: db.find() </pre>	ПК-6
25.		Необходимо выбрать все документы из коллекции cats текущей базы данных. Какую команду консоли следует использовать? <pre> +: db.cats.find() -: db.cats.select() -: db.cats.findAndSelect() -: db.find("cats") </pre>	ПК-6
26.		Необходимо выбрать все документы из коллекции person текущей базы данных. Какую команду консоли следует использовать? <pre> +: db.person.find() -: db.person.select() -: db.person.findAndSelect() -: db.find() </pre>	ПК-6
27.		Необходимо выбрать один документ из коллекции zoo текущей базы данных, удовлетворяющий критерию поиска. Какую команду консоли следует использовать? <pre> +: db.person.findOne({name: "Tiger"}) -: db.person.find({name: "Tiger"}) </pre>	ПК-6

		-: db.person.One({name: "Tiger"}) -: db.person.selectOne({name: "Tiger"})	
28.		Необходимо выбрать один документ из коллекции zoo текущей базы данных, удовлетворяющий критерию поиска. Какую команду консоли следует использовать? +: db.person.findOne({age: 30}) -: db.person.find({age: 30}) -: db.person.One({age: 30}) -: db.person.selectOne({age: 30})	ПК-6
29.		В коллекции goods содержатся документы: {type: "A", name: "Хлеб"} {type: "A", name: "Молоко"} {type: "B", name: "Спрей"} Были выполнены следующие команды: db.goods.insertOne({type: "A", product: "Сыр"}) db.goods.insert([{name: "No name"}, {type: "B", name: "Духи"}]) db.goods.getcount() Что будет выведено в консоль? -: 6 +: Error -: 2 -: 3 -: 1 -: 5 -: 4	ПК-6
		Задания открытого типа	ПК-6
30.	2	В коллекции city содержатся документы: {type: "Village", name: "Woodburn"} {type: "Town", name: "Bookland"} {type: "Town", name: "Prokland"} Сколько документов вернет запрос db.city.find({type: "Town"})?	ПК-6
31.	0	В коллекции places содержатся документы: {type: "Village", name: "Woodburn"} {type: "Town", name: "Bookland"}	ПК-6

		{type: "Town", name: "Prokland"} Сколько документов вернет запрос db.places.find({name: "Town"})?	
32.	1	В коллекции city содержатся документы: {type: "Village", name: "Woodburn"} {type: "Town", name: "Bookland"} {type: "Town", name: "Prokland"} Сколько документов вернет запрос db.city.find({type: "Village"})?	ПК-6
33.	3	В коллекции city содержатся документы: {type: "Village", name: "Woodburn"} {type: "Town", name: "Bookland"} {type: "Town", name: "Prokland"} Сколько документов вернет запрос db.city.find()?	ПК-6
34.	5	В коллекции goods содержатся документы: {type: "A", name: "Хлеб"} {type: "A", name: "Молоко"} {type: "B", name: "Спрей"} {type: "B", name: "Салфетка"} {type: "C", name: "NoName"} Сколько документов вернет запрос db.goods.find()?	ПК-6
35.	2	В коллекции goods содержатся документы: {type: "A", name: "Хлеб"} {type: "A", name: "Молоко"} {type: "B", name: "Спрей"} {type: "B", name: "Салфетка"} {type: "C", name: "NoName"} Сколько документов вернет запрос db.goods.find({ type : "B" })?	ПК-6
36.	1	В коллекции goods содержатся документы: {type: "A", name: "Хлеб"} {type: "A", name: "Молоко"} {type: "B", name: "Спрей"} {type: "B", name: "Салфетка"} {type: "C", name: "NoName"} Сколько документов вернет запрос db.goods.findOne({ type : "B" })?	ПК-6

37.	1	В коллекции goods содержатся документы: <pre>{type: "A", name: "Хлеб"} {type: "A", name: "Молоко"} {type: "B", name: "Спрей"} {type: "B", name: "Салфетка"} {type: "C", name: "NoName"}</pre> Сколько документов вернет запрос <code>db.goods.find({ type : "B" }).limit(1)?</code>	ПК-6
38.	2	В коллекции goods содержатся документы: <pre>{type: "A", name: "Хлеб"} {type: "A", name: "Молоко"} {type: "B", name: "Спрей"} {type: "B", name: "Салфетка"} {type: "C", name: "NoName"}</pre> Сколько документов вернет запрос <code>db.goods.find({ type : "B" }).limit(3)?</code>	ПК-6
39.	2	В коллекции goods содержатся документы: <pre>{type: "A", name: "Хлеб"} {type: "A", name: "Молоко"} {type: "B", name: "Спрей"} {type: "B", name: "Салфетка"} {type: "C", name: "NoName"}</pre> Сколько документов вернет запрос <code>db.goods.find({ type : "B" }).limit(0)?</code>	ПК-6
40.	3	В коллекции goods содержатся документы: <pre>{type: "A", name: "Хлеб"} {type: "A", name: "Молоко"} {type: "B", name: "Спрей"}</pre> Сколько документов вернет запрос <code>db.goods.find().limit(0)?</code>	ПК-6
41.	1	В коллекции goods содержатся документы: <pre>{type: "A", name: "Хлеб"} {type: "A", name: "Молоко"} {type: "B", name: "Спрей"}</pre> Сколько документов вернет запрос <code>db.goods.find({ type : "B" }).limit(2)?</code>	ПК-6
42.	2	В коллекции goods содержатся документы: <pre>{type: "A", name: "Хлеб"}</pre>	ПК-6

		<pre>{type: "A", name: "Молоко"} {type: "B", name: "Спрей"} Сколько документов вернет запрос db.goods.find({ type : "A" }).limit(2)?</pre>	
43.	2	В коллекции goods содержатся документы: <pre>{type: "A", name: "Хлеб"} {type: "A", name: "Молоко"} {type: "B", name: "Спрей"}</pre> Сколько документов вернет запрос db.goods.find({ type : "A" }).limit(0)?	ПК-6
44.	2	В коллекции goods содержатся документы: <pre>{type: "A", name: "Хлеб"} {type: "A", name: "Молоко"} {type: "B", name: "Спрей"}</pre> Что вернет запрос db.goods.find({ type : "A" }).count()?	ПК-6
45.	1	В коллекции goods содержатся документы: <pre>{type: "A", name: "Хлеб"} {type: "A", name: "Молоко"} {type: "B", name: "Спрей"}</pre> Что вернет запрос db.goods.find({ type : "B" }).count()?	ПК-6
46.	6	В коллекции goods содержатся документы: <pre>{type: "A", name: "Хлеб"} {type: "A", name: "Молоко"} {type: "B", name: "Спрей"}</pre> Были выполнены следующие команды: <pre>db.goods.insertOne({type: "A", product: "Сыр"}) db.goods.insert([{name: "No name"}, {type: "B", name: "Спрей"}]) db.goods.find().count()</pre> Что будет выведено в консоль?	ПК-6
47.	2	В коллекции goods содержатся документы: <pre>{type: "A", name: "Хлеб"} {type: "A", name: "Молоко"} {type: "B", name: "Спрей"}</pre> Были выполнены следующие команды: <pre>db.goods.insertOne({type: "A", product: "Сыр"})</pre>	ПК-6

		db.goods.insert ([{name: "No name"}, {type: "B", name: "Духи"}]) db.goods.find({ type: "B"}).count() Что будет выведено в консоль?	
48.	3	В коллекции goods содержатся документы: {type: "A", name: "Хлеб"} {type: "A", name: "Молоко", date : "07.2019"} {type: "B", name: "Спрей"} Были выполнены следующие команды: db.goods.insertOne({type: "A", product: "Сыр"}) db.goods.insert ([{name: "No name"}, {type: "B", name: "Спрей"}]) db.goods.find({ type: "A"}).count() Что будет выведено в консоль?	ПК-6
49.	2	В коллекции goods содержатся документы: { name : "Хлеб" , date : "2020"} {type : "A", name: "Молоко" } {type : "B", name: "Спрей", date : "2019"} {type : "B", name: "Салфетка" , date : "2018"} {type : "C", name: "NoName" , date : "2020"} Сколько документов вернет запрос db.goods.find({ date : "2020" }).limit(0)?	ПК-6
50.	3	В коллекции goods содержатся документы: { name : "Хлеб" , date : "2020"} {type : "A", name: "Молоко" } {type : "B", name: "Спрей", date : "2019"} {type : "B", name: "Салфетка" , date : "2020"} {type : "C", name: "NoName" , date : "2020"} Сколько документов вернет запрос db.goods.find({ date : "2020" }).limit(0)?	ПК-6
51.	1	В коллекции goods содержатся документы: {type: "A", name: "Хлеб"} {type: "A", name: "Молоко"} {type: "B", name: "Спрей"} Были выполнены следующие команды: db.goods.insertOne({type: "A", product: "Сыр"}) db.goods.insert ([{name: "No name"}, {type: "C", name: "Духи"}])	ПК-6

		db.goods.find({ type: "C"}).count() Что будет выведено в консоль?	
52.	2	В коллекции goods содержатся документы: {type: "A", name: "Хлеб"} {type: "A", name: "Молоко"} {type: "B", name: "Спрей"} {type: "B", name: "Молоко"} Были выполнены следующие команды: db.goods.insertOne({type: "A", product: "Сыр"}) db.goods.insert ([{name: "No name"}, {type: "C", name: "Духи"}]) db.goods.find({ name: "Молоко"}).count() Что будет выведено в консоль?	ПК-6

2. Описание шкалы оценивания

В рамках рейтинговой системы успеваемость студентов по каждой дисциплине оценивается в ходе текущего контроля и промежуточной аттестации. Рейтинговая система оценки знаний студентов основана на использовании совокупности контрольных мероприятий по проверке пройденного материала (контрольных точек), оптимально расположенных на всем временном интервале изучения дисциплины. Принципы рейтинговой системы оценки знаний студентов основываются на требованиях, описанных в Положении об организации образовательного процесса на основе рейтинговой системы оценки знаний студентов в ФГАОУ ВО «СКФУ».

3. Критерии оценивания компетенций заданий открытого типа

Оценка **«отлично»** выставляется студенту, если являются верными результаты демонстрации знаний об основных платформах, технологиях и инструментальных программно-аппаратных средствах реализации информационных систем, применения современных технологий для реализации информационных систем, выбора платформ для реализации информационных систем, анализа и выбора инструментальных программно-аппаратных средств для реализации информационных систем; представленный документ не содержит ошибок и решены все задачи. Компетенция ПК-6 освоена на высоком уровне.

Оценка **«хорошо»** выставляется студенту, если являются верными результаты демонстрации знаний об основных платформах, технологиях и инструментальных программно-аппаратных средствах реализации информационных систем, применения современных технологий для реализации информационных систем, выбора платформ для реализации информационных систем, анализа и выбора инструментальных программно-аппаратных средств для реализации информационных систем; отчетный документ содержит незначительные ошибки, не влияющие на общий результат и не представлены отдельные задачи, не влияющие на общий результат. Компетенция ПК-6 освоена на среднем уровне

Оценка **«удовлетворительно»** выставляется студенту, если в основном верными являются результаты демонстрации знаний об основных платформах, технологиях и инструментальных программно-аппаратных средствах реализации информационных систем, применения современных технологий для реализации информационных систем, выбора платформ для реализации информационных систем, анализа и выбора инструментальных программно-аппаратных средств для реализации информационных систем; отчетный документ содержит ошибки, не позволяющие оценить качество разработки; представлены не все данные и задачи решены не в полном объеме. Компетенция ПК-6 освоена на минимальном уровне.

Оценка **«неудовлетворительно»** выставляется студенту, если являются неверными результаты демонстрации знаний об основных платформах, технологиях и инструментальных программно-аппаратных средствах реализации информационных систем, применения современных технологий для реализации информационных систем, выбора платформ для реализации информационных систем, анализа и выбора инструментальных программно-аппаратных средств для реализации информационных систем; представленный документ содержит много грубых ошибок, не представлены ключевые данные и задачи решены недостаточно полно. Компетенция ПК-6 не сформирована.

4. Критерии оценивания компетенций заданий закрытого типа

Оценка **«отлично»** выставляется студенту, если он правильно ответил на 88-100% от общего количества заданий. Компетенция ПК-6 освоена на высоком уровне.

Оценка **«хорошо»** выставляется студенту, если он правильно ответил на 72-87% от общего количества заданий. Компетенция ПК-6 освоена на среднем уровне.

Оценка «удовлетворительно» выставляется студенту, если он правильно ответил на 53-71% от общего количества заданий. Компетенция ПК-6 освоена на минимальном уровне.

Оценка «неудовлетворительно» выставляется студенту, выставляется студенту, если он правильно ответил на менее 53% от общего количества заданий. Компетенция ПК-6 не сформирована.